



**TECNOLÓGICO  
NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO  
DE CUAUTLA**

# Práctica 1

## Flor de Vida

Gálvez Cadenas Oscar | Graficación | 16/02/2026  
24680111 | Cuarto Semestre - Grupo 1 | Ingeniería en Sistemas Computacionales  
Unidad 1 | Profesor. Caballero Alfaro Arístides

# Reporte de Práctica: Geometría Generativa (Flor de la Vida)

Este documento detalla la implementación de un algoritmo en Python para la generación de geometría basada en desplazamientos polares dentro de Blender.

## Fundamentos Matemáticos

Para la disposición de los elementos, se utilizó la conversión de **Coordenadas Polares**  $(r, \theta)$  a **Coordenadas Cartesianas**  $(x, y)$ . Dado que las funciones trigonométricas en Python operan en radianes, la relación utilizada es:

$$x = r \cos(\theta)$$

$$y = r \sin(\theta)$$

Donde  $\theta_{\text{rad}} = \theta_{\text{deg}} \frac{\pi}{180}$

## Lógica del Algoritmo

La práctica se centra en el concepto de **Desplazamiento del Centro**. El flujo de control sigue estos pasos:

1. **Limpieza del Buffer:** Se eliminan objetos preexistentes en la escena para evitar solapamientos.
2. **Instanciación de Referencia:** Se crea un círculo base en el origen  $(0, 0, 0)$ .
3. **Iteración Angular:** Mediante un ciclo `while`, se calculan las posiciones en el perímetro del círculo base cada  $60^\circ$  (división exacta de  $360/6$ ).
4. **Generación de Primitivas:** Se añade un nuevo círculo en cada coordenada calculada.

## Lógica de Construcción: El Desplazamiento del Centro

La construcción de esta figura geométrica sigue una regla de oro: **la ubicación del centro de un círculo depende del radio del círculo anterior.**

### *El Patrón Hexagonal*

Para obtener una simetría perfecta de 6 pétalos (típica de la Flor de la Vida), dividimos la circunferencia total ( $360^\circ$ ) entre el número de círculos deseados:

- **Círculos:** 6
- **Paso angular:**  $360/6 = 60^\circ$

Cada nuevo círculo se desplaza sobre el perímetro del círculo central, creando las intersecciones que forman la figura.

### Desglose del Algoritmo

El proceso de ejecución del script sigue un flujo lógico de cuatro etapas:

1. **Limpieza del Entorno:** Se utiliza `bpy.ops.object.select_all` y `delete` para asegurar que cada vez que corras el script, la escena esté vacía y no se encimen geometrías previas.
2. **Generación del Origen:** Se traza el primer círculo en la coordenada  $(0, 0, 0)$ . Este servirá como nuestra "guía" visual y matemática.
3. **Bucle de Iteración (While):** El script entra en un ciclo que se repite mientras el ángulo actual sea menor a  $360^\circ$ . En cada iteración:
  - Calcula las coordenadas  $(x, y)$  usando el radio y el ángulo actual.
  - Instancia un nuevo círculo en esa posición.
  - Suma  $60^\circ$  al ángulo para preparar la siguiente posición.
4. **Finalización:** Una vez que el ciclo completa los  $360^\circ$ , la figura queda cerrada y perfectamente simétrica.